LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-410446

Implementation of Accelerated Beam-Specific Matched-Filter-Based Optical Alignment

A. A. S. Awwal, K. L. Rice, T. M. Taha

February 9, 2009

Applied Optics

Disclaimer

Implementation of Accelerated Beam-Specific Matched-Filter-Based Optical Alignment

Abdul A. S. Awwal,[1] Kenneth L. Rice,[2] and Tarek M. Taha[2]

[1] *National Ignition Facility*

Lawrence Livermore National Laboratory, Livermore, CA 94551

[2] *Electrical and Computer Engineering Department, Clemson University, Clemson, SC 29631*

[*] *Corresponding author: awwal1@llnl.gov*

Abstract

Accurate automated alignment of laser beams in the National Ignition Facility (NIF) is essential for achieving extreme temperature and pressure required for inertial confinement fusion. The alignment achieved by the integrated control systems relies on algorithms processing video images to determine the position of the laser beam images in real-time. Alignment images that exhibit wide variations in beam quality require a matched-filter algorithm for position detection. One challenge in designing a matched-filter based algorithm is to construct a filter template that is resilient to variations in imaging conditions while guaranteeing accurate position determination. A second challenge is to process the image as fast as possible. This paper describes the development of a new analytical template that captures key recurring features present in the beam image to accurately estimate the beam position under good image quality conditions. Depending on

the features present in a particular beam, the analytical template allows us to create a highly tailored template containing only those selected features. The second objective is achieved by exploiting the parallelism inherent in the algorithm to accelerate processing using parallel hardware that provides significant performance improvement over conventional processors. In particular, a Xilinx Virtex II Pro FPGA hardware implementation processing 32 templates provided a speed increase of about 253 times over an optimized software implementation running on a 2.0 GHz AMD Opteron core.

OCIS codes: 070.0070, 100.0100, 070.5010, 100.2000, 100.3008, 100.4999, 100.5010, 150.5758, 220.1140, 350.4600, 140.3538, 350.266, 140.0140

1.0 INTRODUCTION

The National Ignition Facility, currently under construction at the Lawrence Livermore National Laboratory, is a stadium-sized facility containing a 192-beam, 1.8-megajoule, 500-terawatt, ultraviolet laser system for the study of inertial confinement fusion and the physics of matter at extreme temperatures and pressures [1]. As of December 2008, the facility is over 99% complete and has recently demonstrated 1.25 MJ 1ω (1.053μ wavelength) and 650 kJ-3ω (351 nm) respectively in 111 beams delivered to the target chamber. Automatic alignment (AA) based on computer analysis of fiducials in video images [2-5] enables scientists to direct large numbers of extremely high-energy laser beams with precise alignment to produce temperatures approaching 100 million K and pressures at 100 billion atmospheres, conditions which are designed to achieve ignition of a deuterium-tritium (DT) fusion target [6]. The AA system autonomously operates 35,000 computer-controlled devices, such as motorized actuators and

video cameras, to adjust mirrors and other optics to perform 26 separate optical adjustments on each of the 192 NIF beams in less than 15 minutes [5]. Extremely fast computational hardware will be essential for reducing the alignment time even further for future continuously-operating laser fusion systems [7] that will be capable of virtually inexhaustible and carbon-free energy production.

At the heart of the automatic alignment system is the beam position detection algorithm, which determines the position of beam fiducials from sensor images taken along the laser beam path. Example beam fiducials are Gaussian beams produced by fiber light source and geometric patterns (e.g., small circles or squares) imprinted on optical components. A number of alignment fiducials are utilized to distinguish between various beams, such as reference beams and main beams. Processing the diverse types of sensor images found in the NIF optical system resulted in a suite of twenty-four image processing algorithms. Algorithms need to be robust to work well under varying optical conditions, but the results must always be reliable. Algorithms such as centroids, Hough transforms, templates, and matched filters are responsible for locating fiducial features and while also assessing image quality. The laser optical system may introduce aberrations affecting quality that must be handled. For example, excessive wavefront error degrades quality by distorting all or portions of the image. Algorithms are designed to successfully process many types of degraded images, and the alignment system is more robust in these cases.

For laser beam images, centroiding is an acceptable technique for determining the beam position [8-10]. However, some beam images may exhibit significant intensity variation or other distortions which makes such an approach susceptible to high position uncertainty; in these

cases, correlation [11] or matched filtering results in excellent stability even in the presence of fluctuating intensity [12]. Matched filtering using simple templates can achieve fairly stable position detection despite a wide range of intensity and beam quality variations. However, simple templates may not always lead to sufficiently accurate results. This work demonstrates a template design that yields more accurate results for good beam quality than could be obtained using the simple template, although at the expense of extra processing time required for template creation.

Many of the template-based beam alignment operations are based on 2-D FFT operations which in turn are based on the 1-D FFT. The parallelism inherent in the FFT algorithm allows the hardware implementation to deliver a significant performance improvement over software implementations running on conventional processors. FFT operations allow achieving higher speed by pipelining computations in an FPGA processor. In this study, an FPGA hardware implementation was developed and compared with optimized C and Matlab software implementations. Any compiled Matlab or compiled IDL code which is currently being used in NIF, will only perform as good as a C code but not better. The implementations were tested with a variety of template images. When using 32 template images, the FPGA provided a speedup of about 253 times over the fastest software implementation examined on a 2.0 GHz AMD Opteron core. This FPGA implementation builds on the preliminary designs presented in [13,14], where speedup factors of only 6 to 20 were achieved. Currently, NIF's 192 beams are aligned in approximately 12 minutes for shot cycles lasting from 4 to 8 hours using compiled IDL algorithms. A faster approach, such as FPGA hardware, will be very useful for lasers requiring continuous alignment operation.

2. Background

The matched filtering technique utilizes a given object with a known position as a template to find the position of a second object by detecting its position in the correlation domain. The *classical matched filter* (CMF) [15] and its variation *phase only filter* (POF) [16] are popular methods for detecting the presence of an object in the presence of noise and distortions. The *amplitude modulated phase only filter* (AMPOF) [17,18] was designed to further enhance the performance of the POF by modulating the POF by an inverse type of amplitude.

The matched filter can be elegantly described mathematically in the Fourier domain. Let the Fourier transform of the template function f (x, y) be denoted by:

$$F(U_x, U_y) = \left|F(U_x, U_y)\right| \exp(j\Phi(U_x, U_y)) \tag{1}$$

and that of the input scene *g(x, y)* containing a replica of the template be represented by:

$$G(U_x, U_y) = \left|G(U_x, U_y)\right| \exp(j\Psi(U_x, U_y)) \tag{2}$$

A CMF corresponding to this function f (x, y) which produces its autocorrelation function is given by the complex conjugate of the template Fourier spectrum as denoted by Eq. 3.

$$H_{CMF}(U_x, U_y) = F^*(U_x, U_y) = \left|F(U_x, U_y)\right| \exp(-j\Phi(U_x, U_y)) \tag{3}$$

From the Fourier transform theory of correlation one can show that the inverse Fourier transformation of the product of F (U_x, U_y) and H_{CMF} (U_x, U_y) results in the convolution of f (x,

y) and f (-x, -y), which is the equivalent of the autocorrelation of f (x, y). The cross-correlation of input image and the target is simply:

$$C_{CMF}(\Delta x, \Delta y) = F^{-1}\{G(U_x, U_y)H_{CMF}(U_x, U_y)\} \tag{4}$$

The position of the template in the input scene can be found from the position of the cross-correlation, auto-correlation, and the reference position of the template using Eqs. 5-6.

$$x_{pos} = x_{cross} - x_{auto} + x_c \tag{5}$$

$$y_{pos} = y_{cross} - y_{auto} + y_c \tag{6}$$

where (x_{pos}, y_{pos}) is the to-be-determined position of the pattern in the image plane, (x_{auto}, y_{auto}) is the position of the template autocorrelation peaks and (x_{cross}, y_{cross}) is the position of the cross-correlation peak. The position of the cross-correlation peak was estimated using a polynomial fit to the correlation peak. The center of the template, (x_c, y_c), and (x_{auto}, y_{auto}) are constant and may be calculated off-line, while the cross-correlation peaks move with the object position. If the template is aligned at the center of the image then the image center (x_c, y_c) cancels the auto-correlation position (x_{auto}, y_{auto}) in Eqs. (5-6). The performance of matched filter is further enhanced when the edges of both the image and template are used because edge detection in the image domain performs an equivalent sharpening of the correlation peak. In particular, a Sobel edge operator is used on both the image and the template.

3. Feature selectable template

Twenty six different control loops arranged at different points along the beam path are used to align the beam to different reference locations. At each of these control loops, a video image of the laser beam is acquired and processed to provide the position of the beam to the control system. All alignment image processing undergoes a three step process [20]. Initially, an off-normal detector is used to determine the validity of the image and reject those images that are adversely affected due to various opto-mechanical malfunctions. After validation, normal images are then processed by the position detection algorithm. A post processor evaluates the uncertainty of the position estimation.

One of the most interesting alignment beam images in NIF appears in the transport spatial filter (TSF) Pass 1 as shown in Fig. 1. The automatic alignment algorithm correlates a given template with these images to determine the beam center. The challenge is to find a single template that accurately locates the center of a set of images irrespective of the beam quality.

The sample images in Fig. 1 illustrate a variety of different conditions that impact image quality. These degradations include non-uniform illumination, geometric distortion, defocus, diffraction, noise and other wavefront effects such as multiple reflections.. These factors are challenging for a single template to match perfectly under all conditions. Consequently, the single template may yield unacceptable estimation errors.

Several features are evident from inspection of these pinhole type images. By studying thousands of images taken from different beam lines in the NIF facility, it was concluded that the most prominent beam feature is the outer circular edge. Even when the beam exhibits a fuzzy edge in some specific segment of the beam, the other parts of the beam are intact and well-

defined. Consequently, we chose a template which matches the outer edge of the beam and also is not influenced by most of the other distortions as long as more than half of the outer edge is intact. This gives us a template as shown in Fig. 2, which finds the center of the beams by matching the perimeter edge. In other words, the edge of the template is matched to the edge of the beam image.

This can be problematic when the shape of the beam is significantly distorted similar to the bottom left image of Fig. 1. In this particular image a simple template will find a match in two different locations, one corresponding to the edge of the actual image and one corresponding to the shadow of the image. A second prominent feature in these image types are fiducial lines. While the fiducial lines appear in the real image, they are not present in the shadow image. Thus if we also match the fiducial lines in addition to the edge of the circle, the algorithm may process this additional information and provide an unambiguous match.

To improve the current centroid finding algorithm to handle a broader range of pinhole image quality, we altered the current template (shown in Fig. 2) to add even more feature information. While template construction using only binary data values is typical, we found that algorithm performance could be improved by constructing templates to include analog features expressed by more than 2 gray levels. We enhanced the binary template by adding more line features and levels of gray as shown in the sample images of Fig. 1. What we observed is the beams usually have features including a center, a horizontal line across the center, and left and right diagonals crossing at the center. Lines within the template are generated from measured slope data. All observed features have an intensity gradient. Several combinations of features were tried to determine optimal performance: 1) center only, 2) horizontal line only, 3) left diagonal line only and 4) right diagonal line only. We determined to include all the features and

then tried varying the number of template gray levels from 2 to 7. Performance improved to a point by increasing the number of gray levels, with the best performance obtained by generating the template using six gray levels.

After generating the modified template (shown in Fig. 3), we compared by visual examination of the images the performance of the binary template against the modified analog template to assess the significance of adding template fiducial features. For the comparison, we created a set of filtered images composed of 192 different images taken from various NIF beams to get a good variance for the test. We processed the filtered image set using the binary template and the modified template. We found a 5% improvement over the binary template when applied to the filtered image set.

In most of the test cases, both the simple binary and gray level templates produced equally good results. However, the gray level template produced better results than the binary template when the beam image under test contained fuzzy edges and strong fiducials. The binary template results in these cases were adversely influenced by the fuzzy edges because the match tends to be best for the strongest edges, which ignores the fiducials.. On the other hand, the gray level template gave worse results with beam cases in which the cross-hair fiducials appeared to be fuzzy. Within the low quality beam types each template had an advantage over the other depending on which part of the image was not well defined. Therefore it was decided that both templates needed to be used for poor quality images as described next.

To optimize performance, the NIF implementation allows for either of the two template types to be used depending on the expected beam quality. This selection of a particular template is made off-line after examining typical actual images. In addition, the algorithm has additional options to select the number of image features to match. For example, some images (such as the

one in top right corner of Fig. 1) may only have two prominent features, so the template only contains those two lines. Alternatively, the image could have just the horizontal and one of the diagonal lines. This additional flexibility allows the template to be tuned *a priori* to the expected beam conditions.

Due to variations in magnification and various imaging conditions the filter may not match well to the exact size of the image, which manifests by weak or multiple correlation peaks. Therefore in practice, a set of filters generated over a range of radii are searched in order to find a best match. The position of the center of the circle that yields the highest correlation is chosen as the beam position. Hardware acceleration of this process facilitates a fast search through the various filters.

4. FPGA Acceleration of Image Correlation using multiple filters

The most computationally intensive portion of the image processing is two-dimensional image correlation. For continuous, high-performance alignment operation such as may be required in a laser inertial confinement fusion power plant, faster methods of beam alignment will be necessary [7]. One advantage of the FFT-based correlation is significant parallelism inherent in the computations, thus enabling potential for greater hardware acceleration. We evaluated hardware acceleration by implementing the image correlation computations on an FPGA. The test system utilized was a Cray XD1 reconfigurable supercomputer using an architecture based on AMD Opteron processing cores (2GHz) and Xilinx Virtex II Pro FPGAs. Data communication in the system is maximized by integrating the FPGAs at the operating

systems level and linking them to AMD Opteron processors through a high-bandwidth, low-latency interconnect. In this system, only one FPGA and AMD core was utilized for the testing since the objective was to compare the performance with a single core CPU and scaled as needed. The AMD core sends images to be processed to the FPGA and receives back the location and peak value in the correlation output.

4.1. Hardware Configuration

Fig. 3 presents a system overview of the FPGA implementation. Input data and intermediate values are stored in buffers (shown as the shaded boxes). These are on-chip memories on the FPGA. The inputs to the system, *f(x, y)* and *g(x, y)*, represent the template and source image in Eqs. (1) and (2), respectively. Up to 32 templates can be loaded into the FPGA (in the buffers labeled f0 to f31) and applied to each source image (in the buffers labeled g0 and g1). The two-dimensional Fast Fourier Transforms (FFTs) in Eqs. (1) and (2) are performed using two consecutive one-dimensional FFTs. Similarly, the inverse FFT in Eq. (5) is implemented with two one-dimensional forward FFTs. The FFT units were built using Xilinx-supplied library components. To enable high-throughput computation, the system is pipelined into a pre-phase Sobel filter and four FFT phases as shown in Fig. 3. In this design, the amount of time required for the Sobel filter computation is the same as the amount of time to complete approximately four phases. Therefore, the pre-phase is designed to occur independently of the phase computations. By alternating the two source image buffers (g0 and g1) between being used as input to the Sobel filter unit and as a memory buffer to hold the incoming source image data, the system allows for the Sobel filter computation to overlap the image-template computations. Each phase works on a particular image–template combination. Since the same set of templates

is used for each image, the templates are preloaded in on-chip buffers. This allows high-speed access to the templates that accelerated the system performance. Note the time to load each template onto the FPGA is longer than the pipeline phase computation time. Since each phase requires multiple cycles to compute, two buffers are needed between consecutive phases. For example, in Fig. 3, the upper buffer (mb0) between Phases 1 and 2 holds the output being generated by Phase 1. The lower buffer (mb1) holds the completed output previously generated by Phase 1, for use Phase 2. Switches pipe data to the appropriate buffers. The pre-phase and the four phases in the architecture perform the following functions:

Pre-Phase: The pre-phase consists of applying a Sobel filter to the input image *g(x, y)* to detect the edges. The time for this stage is only seen once because it is overlapped with the computation of the input image *g(x, y)* with the various filters images.

Phase 1: The first one-dimensional FFT for Complex Fourier transform represented by Eqs. (1) and (2) is computed. These two computations can be carried out in parallel. The inputs to this phase are unsigned 8 bit values. Since an 8 bit FFT unit would treat the inputs as signed values, a larger bit width FFT unit is needed. Therefore a 12 bit FFT unit is used in the first phase. The first phase 12 bit FFT outputs are stored in buffers labeled mb0 and mb1 exiting Phase 1.

Phase 2: The second one-dimensional FFT to complete the Complex Fourier transform represented by Eqs. (1) and (2) is computed. As the maximum output value for Phase 1 is 14 bit, a 16 bit FFT unit is used for the second phase. Also part of Eq. (5) is evaluated. Here the output of Eq. (1) is conjugated and multiplied by the output of Eq. (2). An FFT shift operation is executed in parallel with the multiplication in order to center the image. The 40 bit output is stored in a buffer.

Phase 3: The first one-dimensional FFT for the inverse FFT in Eq. (5) is evaluated. Since the inverse FFT is implemented with two 24-bit forward FFT units, they use only the most significant 24 bits of the inputs. This introduces round-off error as the computations take place in the integer domain.

Phase 4: The second one-dimensional FFT for the inverse FFT is Eq. (5) is evaluated here. Pipelined computation of the location of the peak in the output of Eq. (5) (C_{CMF}) is also determined. The absolute value of each location is computed and then compared against previously generated values to determine the peak location. The coordinates and amplitude of the peak along with the amplitude of the four surrounding locations are stored and returned to the processor. The template where the maximum has occurred among the submitted templates to the FPGA is also returned to the processor.

4.2. Hardware performance

The system above was implemented on a Xilinx Virtex II Pro FPGA (part number XCVP50) on a Cray XD1. The FPGA synthesized system ran at 160 MHz. FPGAs contain a certain amount of logic (AND, OR, etc.) and memory (block RAM) on chip. Any design is converted to a circuit that is programmed on the FPGA. Our circuit used 69% of the logic and 75% of the memory on the FPGA. The algorithm was also implemented in Matlab and in C. The latter was developed because it would provide a fair comparison of the FPGA against a software implementation. The C implementation is more optimized than Matlab. The C implementation ran on a 2 GHz AMD Opteron core, while the Matlab implementation was run on a 3 GHz Pentium 4 processor. The optimized FFT library developed by Stefan Gustavson [20] was utilized in the C software implementation. All systems were tested with 64x64 images and 32 templates per image. The

overall runtime of the FPGA system to process an image through 32 templates was about 0.4 ms, while the C system required 108 ms. The Matlab implementation required 178 ms and compiled Matlab was faster at 141ms. This indicates that the FPGA system provided a speedup of approximately 254 times over the C system and 416 times over the Matlab implementation. Newer generation FPGAs with larger resources and higher clock speeds would allow multiple pipelines to analyze more images in parallel, thus resulting in greater speedups.

The system was tested with the sample and template images shown in Fig. 4. The sample had a diameter of 40 pixels while five versions of the templates were generated with varying diameters (of 32, 37, 41, 45, and 49 pixels). The absolute of the peak values from correlating the sample with the fifteen template images on the FPGA are shown in Table 1. As expected, template 1 produced higher peaks as it is more similar to the sample image than the other templates. The highest peak was obtained for the version of template 1 with a diameter of 41 as this is the closest match to the sample. Table 2 lists the sample image centroid calculated based on the FPGA and Matlab implementation outputs. The results show the numerical FPGA output is very close to the Matlab output despite the rounding error caused by truncating the intermediate results that was necessary in the FPGA.

5. Summary

This paper describes the design of a template flexible enough to be used with corner cube reflected pinhole images of any image quality. Depending on the expected image quality, various template features are selected off-line and remain static during operation. For good quality images, all features are used to give the best performance. Under less optimal image quality

conditions line features also degrade, such that the simple template shown in Fig. 2 is used to avoid an erroneous position estimation.

The parallelism inherent in the matching algorithm allows a hardware implementation to yield significant speedups over software implementations using conventional processors. In this study, a Xilinx Virtex II Pro FPGA hardware implementation was developed and compared with Matlab and optimized C software implementations. The systems were tested with a variety of real-world templates and beam images. When applied to a set of 32 template images such as might be used in an optimal template search, the FPGA provided a speedup of about 253 times over the fastest software implementation examined. This indicates that the hardware acceleration of this algorithm can provide significant speedups.

Acknowledgement

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Abdul Awwal acknowledges the improvements suggested by Paul Van Arsdall. Kenneth Rice acknowledges the summer student support at Lawrence Livermore National Laboratory. Kenneth Rice and Tarek Taha acknowledge grants from the Air Force Research Laboratory (including the AFRL Information Directorate) and a National Science Foundation CAREER award. This work was also supported in part by a grant of computer time from the DOD High Performance Computing Modernization Program at the Naval Research Laboratory.

References

1. E. I. Moses, and C. R. Wuest, "The National Ignition Facility: Laser performance and first experiments." *Fusion Science and Technology* **47**, 314-322, (2005).
2. E. Bliss, M Summers, F. Holloway, B. Johnson, "Shiva Alignment Systems," Journal of The Optical Society of America **68**, 547-547, (1978).
3. E. Stout, V. Miller-Kamm, J. Spann, and P. Van Arsdall, "Prototype distributed object-oriented architecture for image-based automatic laser alignment," Proc. SPIE **3047**, 214-221 (1997).
4. R. A. Zacharias, N. R. Beer, E. S. Bliss, et al., "Alignment and wavefront control systems of the National Ignition Facility," Optical Engineering **43**, 2873-2884 (2004).
5. K. Wilhelmsen, A. Awwal, W. Ferguson, B Horowitz, V. Miller Kamm, C. Reynolds, "Automatic Alignment System For The National Ignition Facility", Proceedings of 2007 International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS07), 486-490, Knoxville, Tennessee, (2007). http://accelconf.web.cern.ch/accelconf/ica07/PAPERS/ROAA02.PDF
6. C. A. Haynam, P. J. Wegner et al., "National Ignition Facility laser performance status," Applied Optics **46**, 3276-3303, (2007).
7. "LIFE: Clean energy from nuclear waste" NIF website https://lasers.llnl.gov/missions/energy_for_the_future/life/
8. R. Irwan and R. G. Lane, "Analysis of optimal centroid estimation applied to Shack_hartmann sensing," Applied Optics **32**, 6737-6743, (1999).
9. Z. Jiang, S. Gong, Y. Dang, "Numerical study of centroid detection accuracy for Shack-Hartmann wavefront sensor," Optics and Laser Technology **38**, 614-619, (2006).

10. J. Arines and J. Ares, "Minimum variance centroid thresholding," Optics Letters **27**, 497-499, (2002).

11. S. Chang and C. P. Grover, "Centroid detection based on optical correlation," Optical Engineering **41**, 2479-2486, (2002).

12. A. A. S. Awwal, Wilbert A. McClay, Walter S. Ferguson, James V. Candy, Thad Salmon, and Paul Wegner, "Detection and Tracking of the Back-Reflection of KDP Images in the presence or absence of a Phase mask," Applied Optics **45**, 3038-3048, (2006).

13. A. A.S. Awwal, K. L. Rice, and T. M. Taha, "Fast implementation of matched-filter-based automatic alignment image processing," *Optics & Laser Technology* **41**, 193-197, (2009).

14. A. A. S. Awwal, R. Leach, K. L. Rice and T. M. Taha, "Higher accuracy template for corner cube reflected image", in Optics and Photonics for Information Processing II edited by Abdul A. S. Awwal, Khan M. Iftekharuddin, Bahram Javidi, Proc. SPIE Vol. **7072** (SPIE Bellingham, WA, 2008) 70720V.

15. VanderLugt, "Signal Detection by Complex Spatial Filtering," IEEE Trans. Inf. Theory **IT-10**, 139-145, (1964).

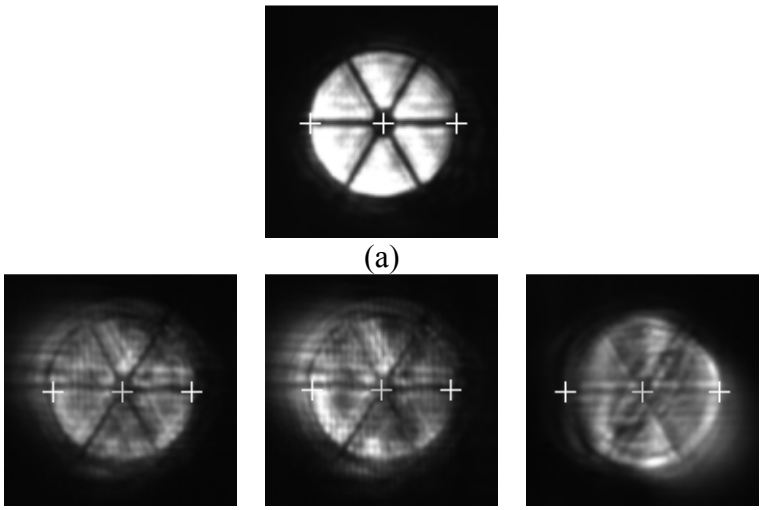
16. J. L. Horner and J. Leger, "Pattern recognition with binary phase-only filters," *Applied Optics* **24**, 609-611, (1985).

17. A. A. S. Awwal, M. A. Karim, and S. R. Jahan, "Improved Correlation Discrimination Using an Amplitude-modulated Phase-only Filter," *Applied Optics* **29**, 233-236, (1990).

18. M. A. Karim and A. A. S. Awwal, *Optical Computing: An Introduction*, John Wiley, New York, NY, 1992.

19. A. Awwal et al., S. W. Ferguson, and C. Law "Uncertainty Detection for NIF Normal Pointing Images," in Optics and Photonics for Information Processing, edited by Abdul A. S. Awwal, Khan M. Iftekharuddin, Bahram Javidi, Proc. SPIE **6695,** 66950R, (2007).

20. FFT code and related material, <http://www.jjj.de/fft/fftpage.html>.



(a)

(b)

Figure 1. (a) Good quality image (b) Fuzzy images

(a) (b)

Figure 2. (a) Binary template for fuzzy images (b) Gray level template for good quality images

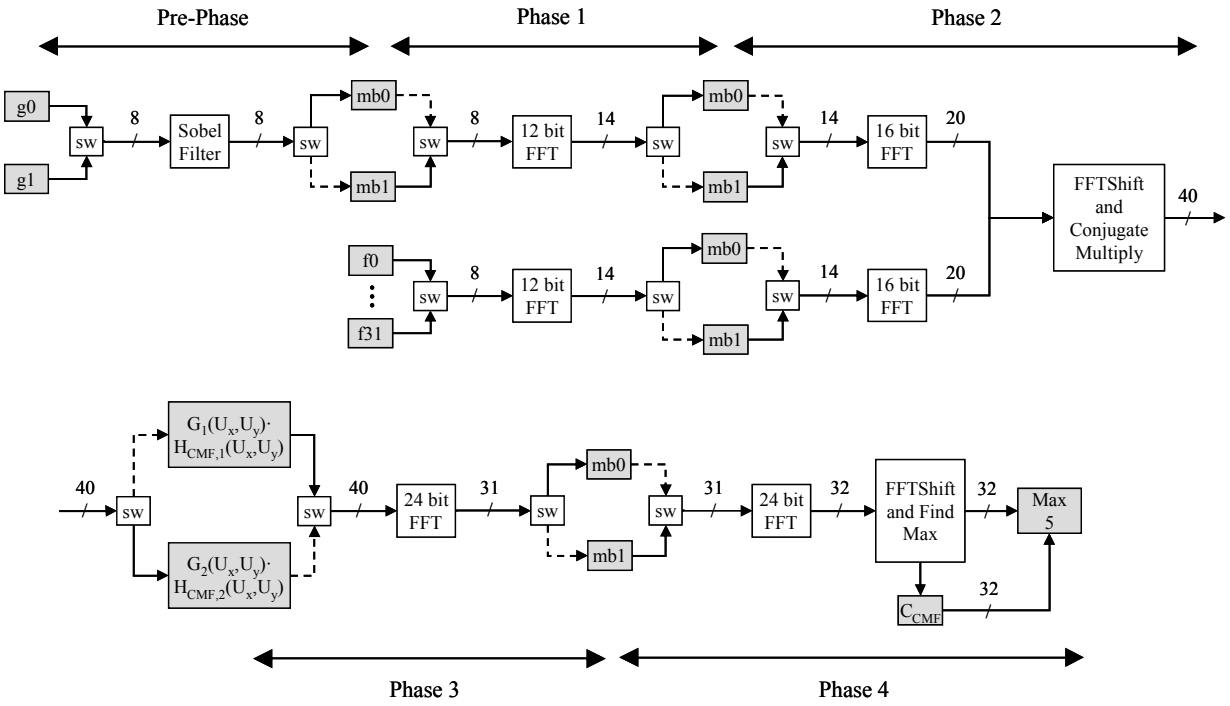


Fig. 3. The block diagram of the FPGA operations. The boxes labeled “sw” are switches.

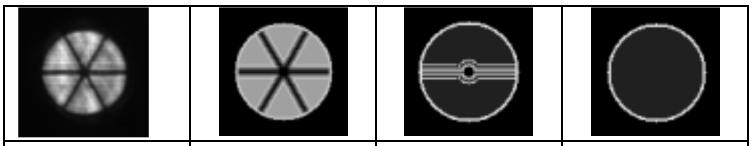


Fig. 4. Sample and template images used for testing. The sample has a diameter of 40 pixels. Five versions of each template were generated with diameters of 32, 37, 41, 45, and 49 pixels.

Table 1. FPGA output of absolute value of peak ($\times 10^9$) in the correlation of the sample and template images shown in Fig. 4. Five versions of each template were used. Template 1 with a diameter of 41 pixels produces the highest peak.

Template diameter	**32**	**37**	**41**	**45**	**49**
Template 1	4.111	6.012	**7.085**	5.661	5.367
Template 2	3.423	4.412	4.922	3.766	3.666
Template 3	2.580	3.776	4.376	2.805	2.689

Table 2. Sample image centroid location calculated based on FPGA and Matlab outputs (using template 1 with diameter 41 pixels).

	x_{cross}	y_{cross}
Matlab	33.169	31.858
FPGA	33.141	32.352
Error across image	-0.043%	0.772%